



**Barcelona
Supercomputing
Center**
Centro Nacional de Supercomputación



Providentia Training Session

Amalia Vradi

Dene Bowdalo

Wednesday, March 24th

What is Providentia?

Providentia is a visualisation tool designed to allow on-the-fly analysis of BSC experiment output, with respect to GHOST or non-GHOST processed observational data.

Data Selection

☐ Colocate

READ

QA

Filters

RESET

Bounds

EEA_AQ_

gas

20180101

FLAGS

hourly

sconco3

20181201

EXPS

% REP

PERIOD

0.0

META

FILTER

400.0

Map Z

Mean

Exp. Bias

Aggregate

Mean

Site Select

☐ All
☒ Intersect

↓

📄

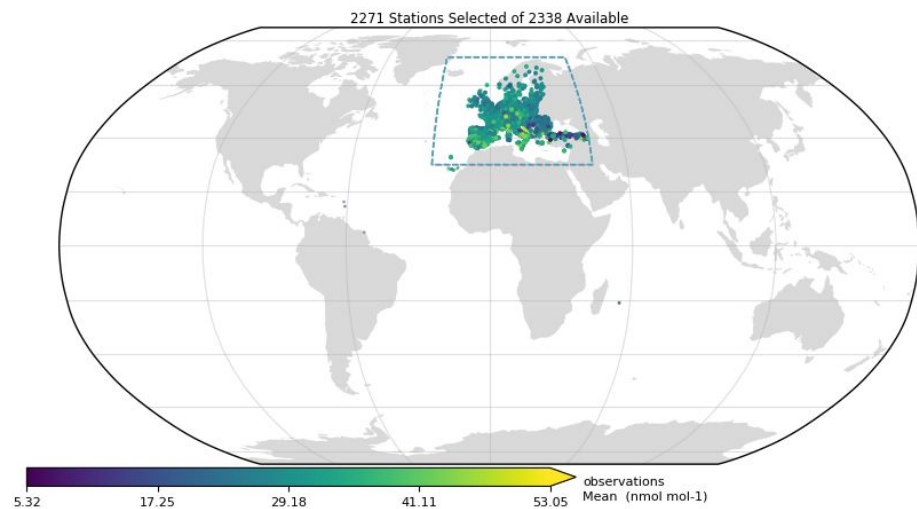
🏠

←

→

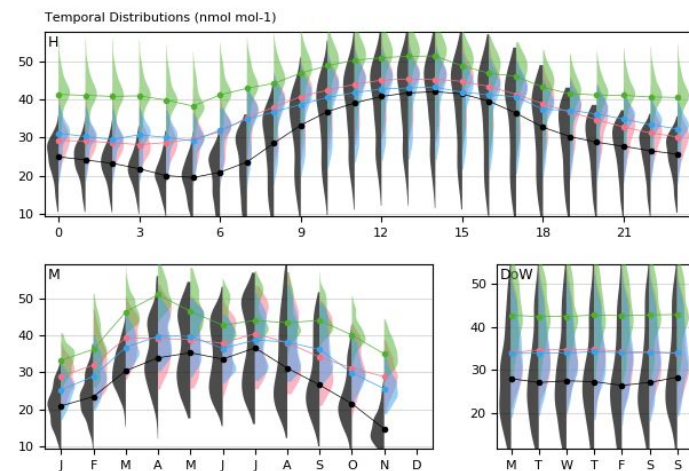
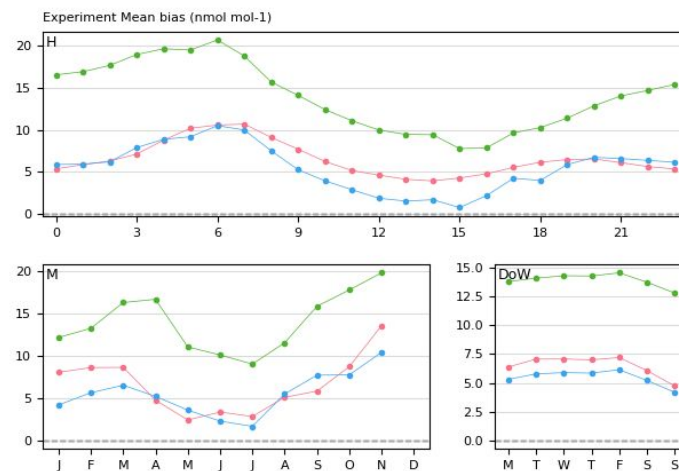
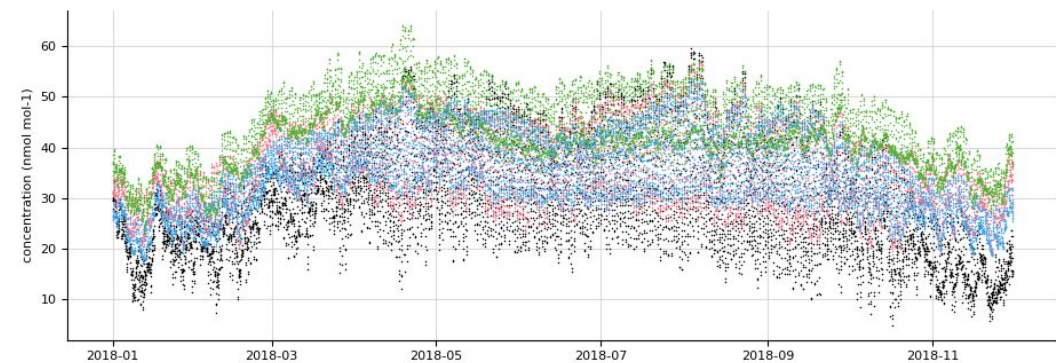
🔍

📁



2271 Stations Selected
 Median Measurement Altitude: 153.60m Median To Coast: -100.00km
 Median Population Density: 2307.5 people/km-2
 Median Nighttime Lights: 56.0
 Country: 39 unique elements
 Area Classification: 7 unique elements
 Station Classification: background:79.0%, point_source-industrial:10.7%, point_source-traffic:7.2%
 Terrain: complex:69.9%, mountain:0.2%, nan:29.8%
 Land Use: 22 unique elements
 MODIS Land Use: 14 unique elements
 GHSL Settlement Class: 9 unique elements
 Measurement Method: differential optical absorption spectroscopy:0.7%, nan:3.0%, ultraviolet photometry:96.3%, unknown:0.0%
 Measuring Instrument: 30 unique elements

● observations ● cams61_chimere_ph2-eu-000 ● cams61_dehm_ph2-eu-000 ● cams61_emep_ph2-eu-000



Steps for using Providentia

Basic workflow:

1. Setup your connection and clone the projects:
 - i. Providentia
 - ii. Providentia Interpolation
2. Interpolate your experiments using Providentia-Interpolation
3. Launch Providentia Dashboard

Extra (Pro 🕶️):

1. Export & Load from your configuration
2. Export data used during your evaluation

Connection Setup



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

SSH Setup

In your local machine, open the config file of SSH:
\$ vi ~/.ssh/config

```
Host mn_1
    HostName mn1.bsc.es
    User bscXXXXX
    IdentityFile ~/.ssh/id_rsa
    ForwardX11Trusted yes
    ForwardX11 yes
    Compression yes
    Ciphers aes128-gcm@openssh.com
    ForwardX11Timeout 7d
```

```
Host nord
    HostName nord1.bsc.es
    User bscXXXXX
    IdentityFile ~/.ssh/id_rsa
    ForwardX11Trusted yes
    ForwardX11 yes
    Compression yes
    ForwardX11Timeout 7d
```

Note that the options for Nord3 **do not** include the line with Ciphers.

Wiki: [Connection Setup](#)

Providentia Interpolation



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

Cloning from GitLab

1. Enter the project's GitLab page: <https://earth.bsc.es/gitlab/ac/providentia-interpolation>
2. In **Clone** copy link from **Clone with HTTPS**
3. In your terminal, clone using the `--recurse-submodules`

```
$ git clone --recurse-submodules https://earth.bsc.es/gitlab/ac/providentia-interpolation.git
```
1. Transfer the folder to the path you usually work. (e.g. your gpfs scratch)

```
$ scp -r providentia-interpolation bsc32XXX@dt01.bsc.es:/gpfs/scratch/bsc32/bsc32XXX/
```


Configure Interpolation I

File:

<https://earth.bsc.es/gitlab/ac/providentia-interpolation/-/blob/master/configuration.py>

- `qos = 'default'`
- `GHOST_version = 'default'`
- `n_neighbours_to_find = 'default'`
- `start_date = '201801' # YYYYMM START FROM THIS POINT`
- `end_date = '201802' # YYYYMM GO UP TO THIS POINT`
- `experiments_to_process = ['cams61_emep_ph2']`
- `species_to_process = ['sconco3']`
- `grid_types_to_process = ['default']`
- `ensemble_options = ['default']`
- `networks_to_interpolate_against = ['EBAS']`
- `temporal_resolutions_to_output = ['hourly']`

Wiki: [Experiment Output Interpolation](#)

[Spreadsheet of participants' expids](#)

Configure Interpolation II

File:

https://earth.bsc.es/gitlab/ac/providentia-interpolation/-/blob/master/defined_experiments.py

Make sure that your experiment is included in the file **defined_experiments.py**

In this file you will find a python dictionary. In the case that your expid is not included, you can add it following the syntax of python and the format of the existing experiments.

```
defined_experiments_dictionary = {# -----
    # MONARCH experiments
    'b007': {'esarchive': '/esarchive/exp/nmmb-bsc-ctm/b007/'},
    'alwd': {'esarchive': '/esarchive/exp/monarch/alwd/'},
    'alvv': {'esarchive': '/esarchive/exp/monarch/alvv/'},
    'alvw': {'esarchive': '/esarchive/exp/monarch/alvw/'},
    'alx8': {'esarchive': '/esarchive/exp/monarch/alx8/'},
    'alxa': {'esarchive': '/esarchive/exp/monarch/alxa/'},
    'alxf': {'esarchive': '/esarchive/exp/monarch/alxf/'},
    'alyq': {'esarchive': '/esarchive/exp/monarch/alyq/'},
    'alyz': {'esarchive': '/esarchive/exp/monarch/alyz/'},
    'alzxc': {'esarchive': '/esarchive/exp/monarch/alzxc/'},
```

Submit Interpolation job

Nord3:

```
$ source load_modules.sh  
$ bsub < n3_experiment_interpolation_submit.sh
```

OR in case of MN4:

```
$ source load_modules.sh  
$ sbatch experiment_interpolation_submit.sh
```

Wiki: [Experiment Output Interpolation](#)

Check Outputs

- Under folder **management_logs** should be written whether the job has finished successfully or not
- The outputs of the interpolation are written in the following path:
 - /gpfs/projects/bsc32/AC_cache/recon/exp_interp
 - The outputs are organized according to the version of GHOST

Providentia Dashboard



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

Cloning from GitLab

1. Enter the project's GitLab page: <https://earth.bsc.es/gitlab/ac/Providentia>
2. In **Clone** copy link from **Clone with HTTPS**
3. In your terminal, clone the repo:

```
$ git clone https://earth.bsc.es/gitlab/ac/Providentia.git
```
1. When we clone a repo in that way, we are by default in the **master branch**. Change to the production:

```
$ cd Providentia
```

```
$ git checkout production
```
1. Transfer the folder to the path you usually work. (e.g. your gpfs scratch)

```
$ scp -r Providentia bsc32XXX@dt01.bsc.es:/gpfs/scratch/bsc32/bsc32XXX/
```

The production branch contains the latest stable release of Providentia

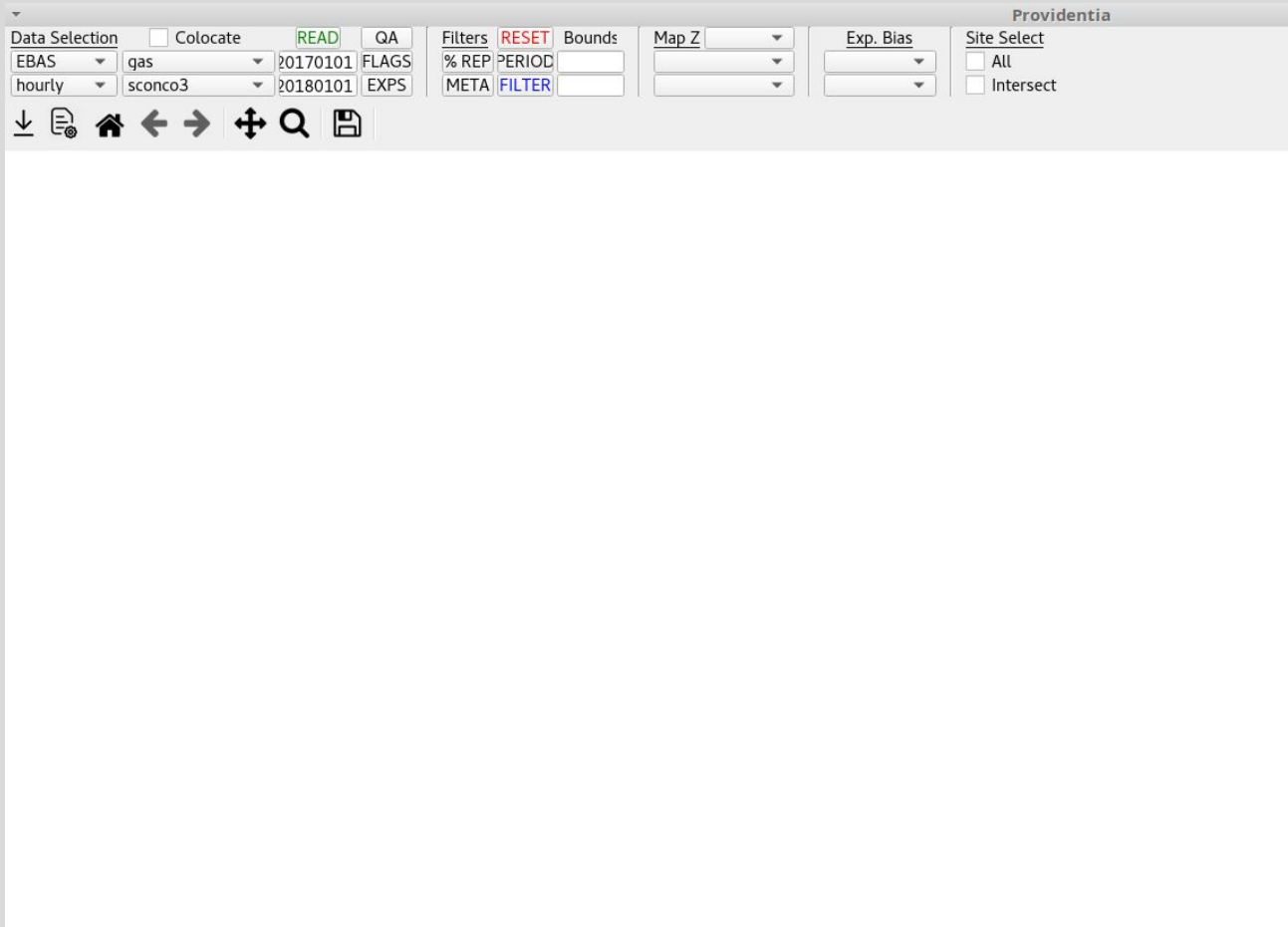
Launch Providentia

In the folder of Providentia:

```
$ source load_modules.sh
```

```
$ ./bin/providentia
```

These commands will load the necessary modules and will request the allocation in the machine we are working (either MN4 or Nord3). When we are granted the allocation, then the dashboard of Providentia pops-up.



When the dashboard window has appeared, it completely blank. We will select the observations and experiments that we want to use for our evaluation.

After having filled our configurations in the menu bar, we will proceed to **READ**.

We will then have our data and experiments loaded and we can see the map, time series and various statistics. That would be the time to select some Metadata filters.

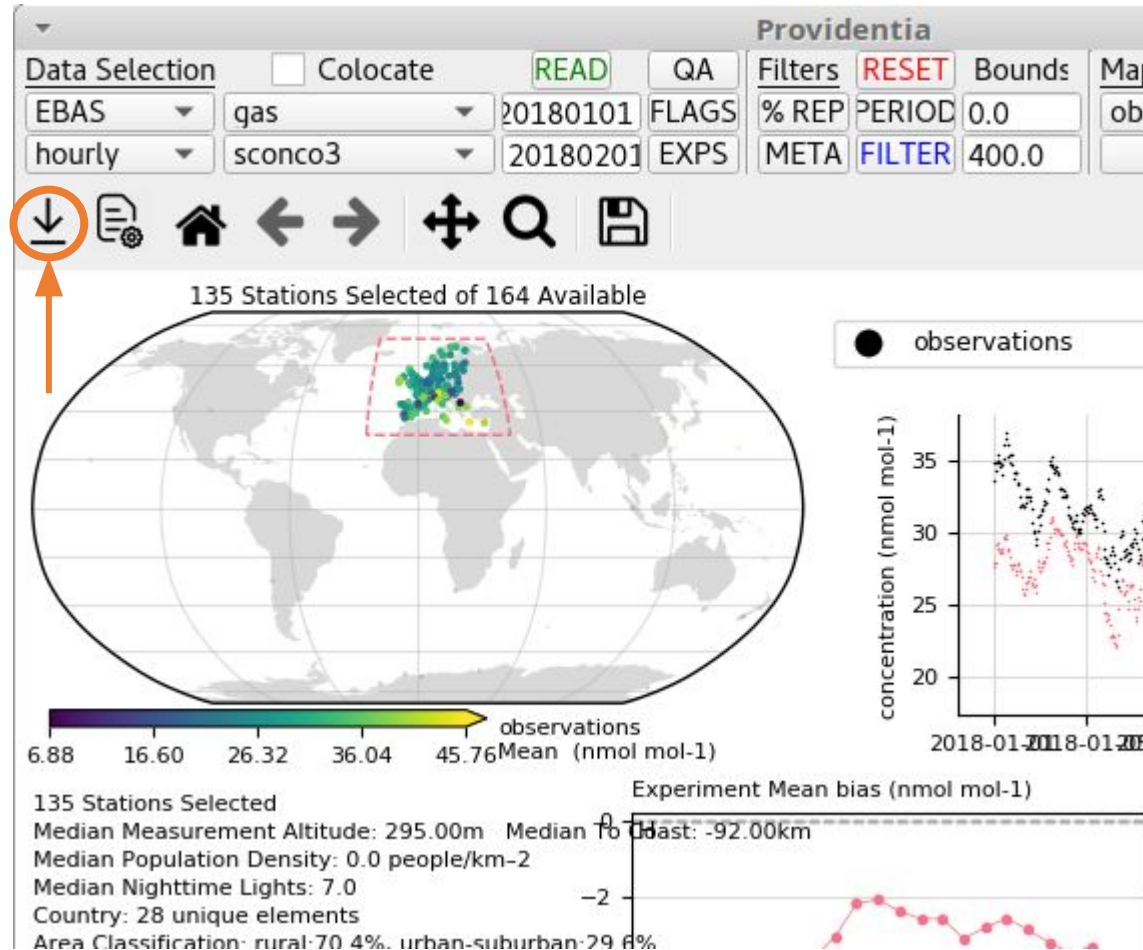
Configuration files



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

Export data or configuration



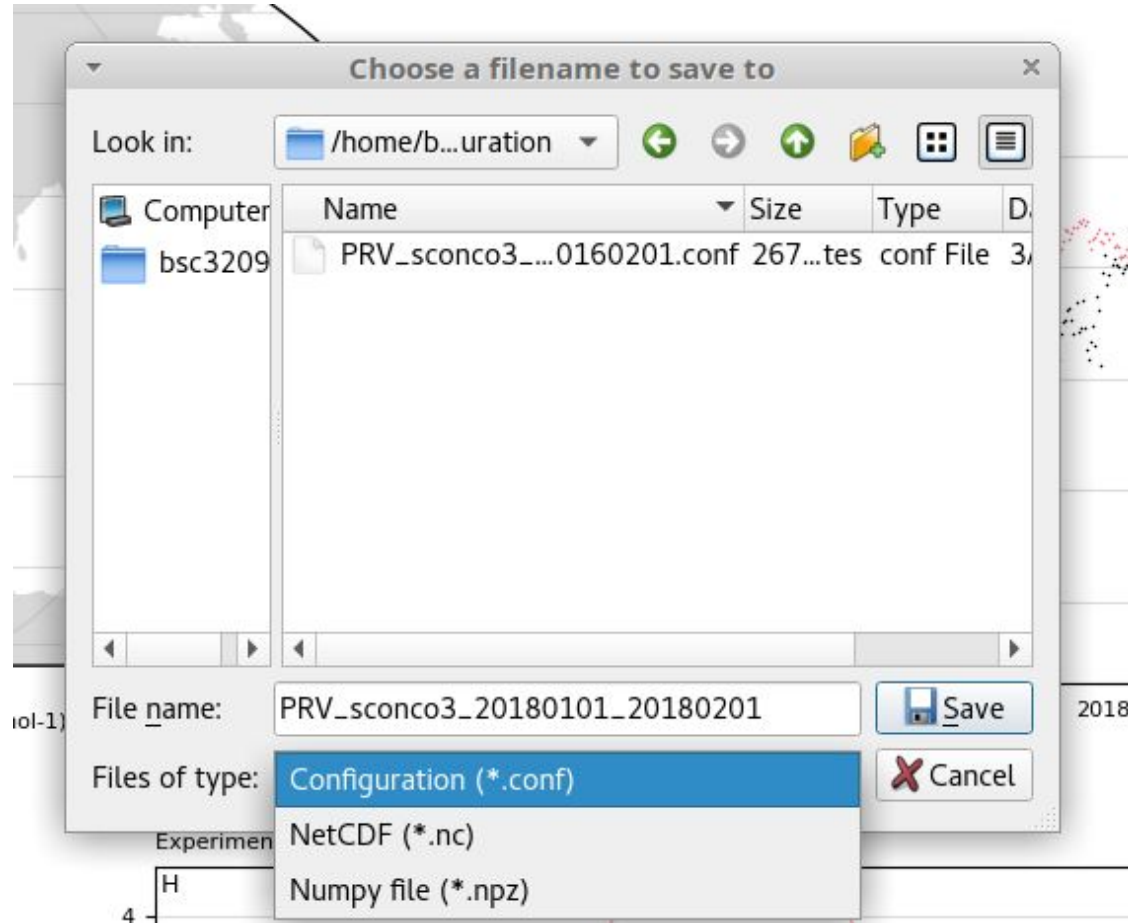
With this option from the toolbar you can:

- Export the data that were used for your evaluation (observations and experiments)
- Export all the configuration options that were used during the evaluation

You can select what you want to export in the dialog that it appears when you select it, along with the filename and the destination.

Wiki pages: [User Guide](#) & [Configuration Files](#)

Export configuration



When we select the option presented previously, we see the dialog of the left. At the bottom, if we select the drop-down menu of **Files of type**, we can select the output that we wish. For this example, we select **Configuration (*.conf)**.

The configuration file is giving a suggested name of the format *PRV_species_start_end*. We can change the name as well as select its destination path.

Wiki pages: [User Guide](#) & [Configuration Files](#)

Format of configuration files

```
[PRV_sconco3_20180101_20180201]
selected_network = EBAS
selected_resolution = hourly
selected_matrix = gas
selected_species = sconco3
start_date = 20180101
end_date = 20180201
experiments = cams61_emep_ph2-eu-000
upper_bound = 35
latitude = -89.9969482422, 50
sampling_height = 0.0, 100
station_classification = keep: background; remove: ;
```

Section name

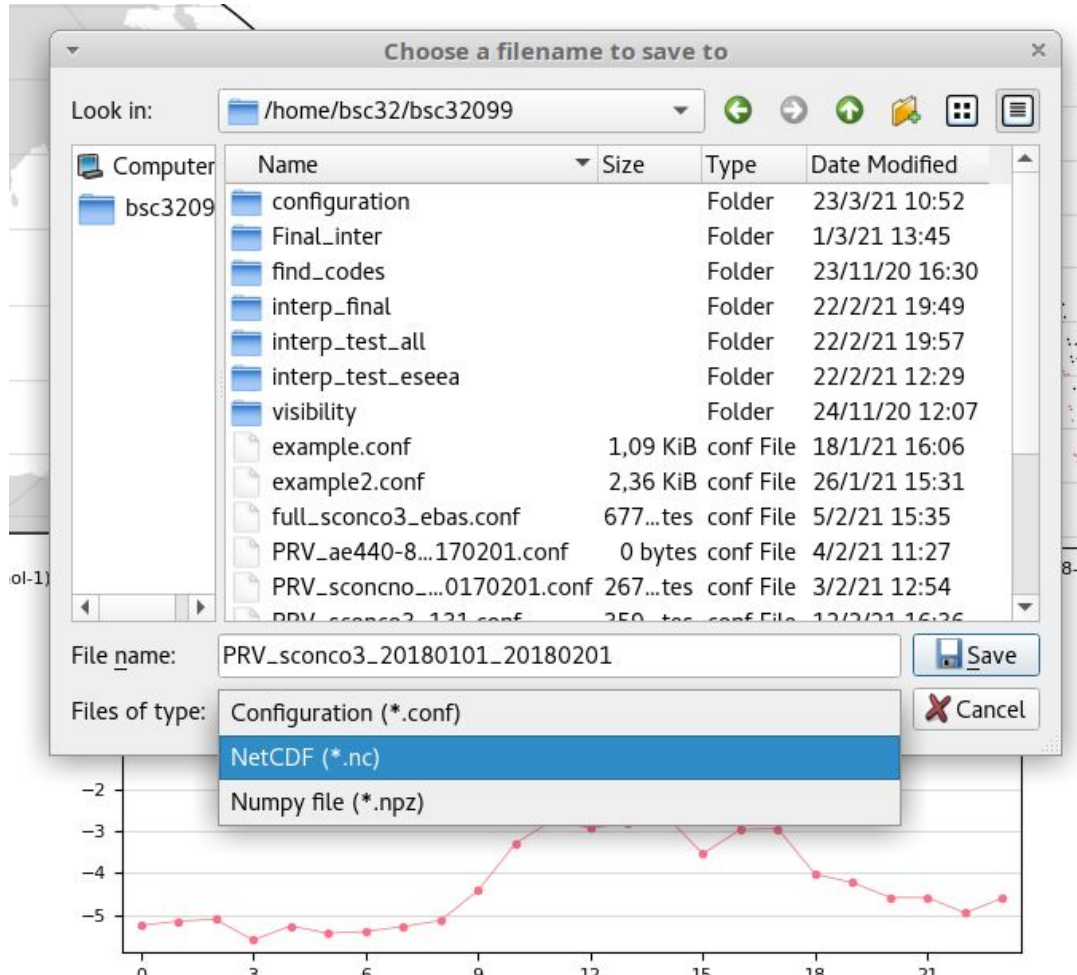
Compulsory fields

Wiki pages: [User Guide](#) & [Configuration Files](#)

Export data

Similarly as before, if we wish to export the data that we used during our evaluation session, we can do so from the menu **File of type**. We can export the data in two format: **netCDF** or **Numpy file**.

The configuration file is giving a suggested name of the format *PRV_species_start_end*. We can change the name as well as select its destination path.



Wiki pages: [User Guide](#) & [Configuration Files](#) & [Saved file formats](#)

Launch Providentia using your configuration

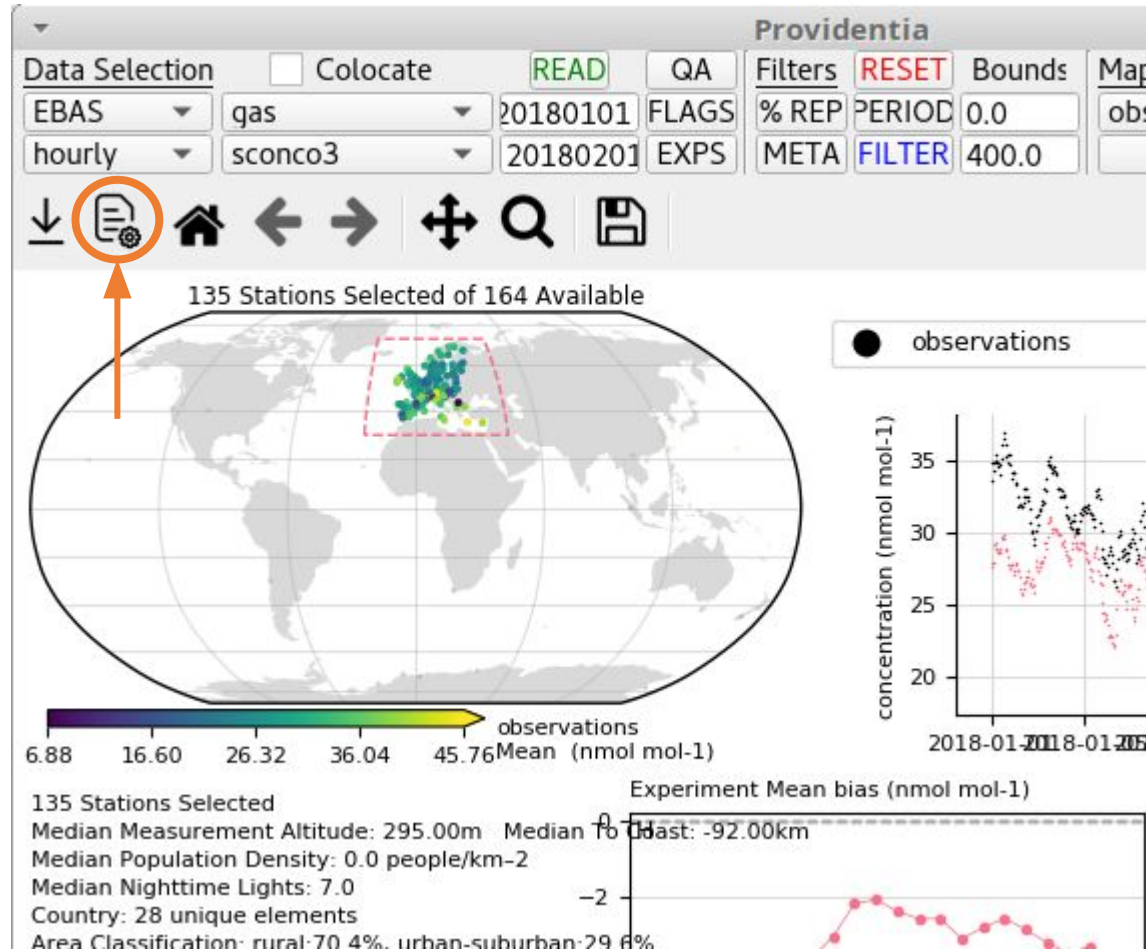
Start the dashboard using your configuration file using the command line:

```
$ ./bin/providentia --config="/path/of/conf/file" --section="section_name"
```

As a configuration file can have multiple, it is necessary to use both command line options of **--config** as well as **--section**.

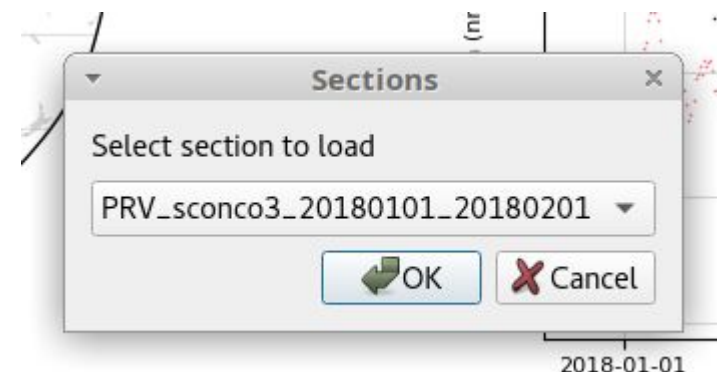
Launching the dashboard in that way, will load the data, experiments and optional filters we have included in the configuration file.

Load configuration from the dashboard



With the option that we show from the toolbar you can load a configuration file. You can select the file from the dialog that will appear.

After selecting the file, an additional dialog will appear asking you to select the section of the configuration that you wish to load. Screenshot of that dialog is presented below.



Wiki pages: [User Guide](#) & [Configuration Files](#)

Hands-on



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

Dashboard Features



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

Test Filtering

Network: **EBAS**, resolution: **hourly**, parameter: **sconco3**,
startdate: **20170101**, enddate: **20180101**

177 Stations

1. all_representativity_percent = **70%**, latitude = **50.0**, measurement_altitude max = **500.0**, land_use keep = **forest**
2. upper bounds = **100**, period keep = **summer & winter**, measuring_instrument_name keep = **Thermo 49i**
3. period remove: **nighttime**, country remove: **France, Germany, United Kingdom**, area_classification keep = **rural**, station_classification keep = **background**
4. lower bounds = **60**, upper bounds = **120**, altitude max = **500.0**, GSFC_coastline_proximity min = **-100**, Meybeck landform classification keep = **hills, lowlands, plains**, GHSL_population_density max = **1000.0**

Test Filtering

Network: **EBAS**, resolution: **hourly**, parameter: **sconco3**,
startdate: **20170101**, enddate: **20180101**

177 Stations

1. all_representativity_percent = **70%**, latitude = **50.0**, measurement_altitude max = **500.0**, land_use keep = **forest** **4 stations**
2. upper bounds = **100**, period keep = **summer & winter**, measuring_instrument_name keep = **Thermo 49i** **35 stations**
3. period remove: **nighttime**, country remove: **France, Germany, United Kingdom**, area_classification keep = **rural**, station_classification keep = **background** **105 stations**
4. lower bounds = **60**, upper bounds = **120**, altitude max = **500.0**, GSFC_coastline_proximity min = **-100**, Meybeck landform classification keep = **hills, lowlands, plains**, GHSL_population_density max = **1000.0** **36 stations**

Specific Usecases



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

non-GHOST files I

All the examples we followed previously were using observations files that had been formatted according to GHOST. If we want to use Providentia using non-GHOST files (e.g. esarchive files) then there are a couple of things we need to pay attention to:

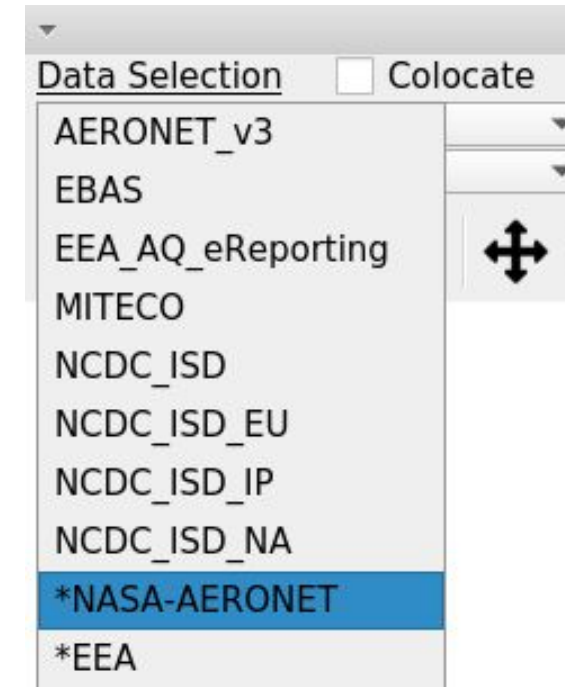
1. in the configuration of Providentia-Interpolation, when we define the variable `networks_to_interpolate_against`, we write the two following folders after `/esarchive/obs/` that correspond to the provider
 - e.g. `nilu/ebas`, `eea/eionet`, `nasa-aeronet/oneill_v3-lev15`
2. For launching the dashboard, we need to run from Nord3, as the alternative of MN4 does not have access to esarchive.
3. In the dashboard, the list of observation providers with esarchive's regular format begin with an asterisk, e.g. `*EEA`.

non-GHOST files II

- `qos = 'default'`
- `GHOST_version = 'default'`
- `n_neighbours_to_find = 'default'`
- `start_date = '201801'`
- `end_date = '201802'`
- `experiments_to_process = ['b012']`
- `species_to_process = ['sconcn02']`
- `grid_types_to_process = ['default']`
- `ensemble_options = ['default']`
- `networks_to_interpolate_against = ['eea/eionet']`
- `temporal_resolutions_to_output = ['hourly']`



configuration of interpolation for non-ghost files



names of esarchive observations begin with asterisk

Ensemble Experiments I

In the latest version of Providentia and its Interpolation, we can interpolate experiments with ensemble members. For interpolating the output of the experiments, the interpolation configuration has the variable **ensemble_options**. The option that this variable can take are listed below:

1. 'allmembers'
2. Number of the statistic, e.g. '000'
3. For ensemble statistics simply define them as 'stat_' + the stat name provided in the filename, e.g. 'stat_av' for ensemble average, 'stat_av_an' for ensemble analysis average, or 'stat_av_inc' for ensemble analysis - first guess average
4. The option 'default' is used for non-ensemble experiments

Ensemble Experiments II

- `qos = 'default'`
- `GHOST_version = 'default'`
- `n_neighbours_to_find = 'default'`
- `start_date = '201801'`
- `end_date = '201802'`
- `experiments_to_process = ['a3bv']`
- `species_to_process = ['sconco3']`
- `grid_types_to_process = ['default']`
- `ensemble_options = ['allmembers']`
- `networks_to_interpolate_against = ['EBAS']`
- `temporal_resolutions_to_output = ['hourly']`

Select Experiment/s

a2yb-regional-000
a2yu-regional-000
a2zw-regional-000
a363-regional-000
a3bv-regional-000
a3bv-regional-001
a3bv-regional-002
a3bv-regional-003
a3bv-regional-004
a3bv-regional-005
a3bv-regional-006
a3bv-regional-007
a3bv-regional-008
a3bv-regional-009
a3bv-regional-010
a3bv-regional-011

☐
☐
☐
☐
☐
☐
☐
☐
☐
☐
☐
☐
☐
☐
☐
☐

Observations and Experiment with different time resolution

When wanting to compare observations with experiment output it is possible that the observations and experiment temporal resolutions are not equivalent, e.g model resolution = **3hourly** and observational resolution = **daily**.

In these instances the model resolution can be resampled to match the observational resolution before performing interpolation. The variable: `temporal_resolutions_to_output` controls the resample resolution. Setting it to **daily** in this instance will resample the model resolution to be daily and comparable with the observations.